



OSW / appiChar Technical Guide for IT Staff

Section 6: Unix

Although systems based on Microsoft's Windows operating system are currently the most common in a majority of offices, Unix-based (and often more specifically Linux-based) systems may also be found lurking in various corners.

Contents:

- Introduction to Unix
- Unix Filesystems
- Useful Commands
- The Visual Editor



Introduction

Unix has its origins in the early 1970s, evolving at the hands of a number of developers within a company called AT&T. They had created a test filesystem and needed a rudimentary operating system to test it with. As the basic system was developed, copies were released to academic institutions and some made significant modifications to the code.

One significant development was the Berkeley Software Distribution (BSD) version which became dominant in the academic and engineering arenas. Most versions of Unix required particular hardware or machines to run on, so that a SUN version of Unix would not run on a Hewlett Packard machine. However, these different versions of Unix are very similar at the user level; it is only when someone needs to administer a Unix system that they should pay particular attention to the version of Unix that they are using. The first significant version of Unix for the PC was Xenix, a commercial product developed by none other than Microsoft.

Currently the most common distributions for use on PCs are Linux and BSD (including FreeBSD and OpenBSD). Each of these is available as a number of different distributions – packages which are aimed at particular platforms or with particular sets of applications included and usually their own customised installation and management facilities.



i	The Unix equivalent of the Windows Administrator user is called root
---	---

Depending on where the Unix machine is and how it has been configured there may be a couple of options for how you use it. You should be able to log in directly with the screen and keyboard attached to the computer. Alternatively you may be able to use telnet from another machine on the network.

Be aware that telnet traffic is not encrypted as it travels over the network so it is possible that it could be “sniffed” and someone else could find out the user name and password that you are using. A better alternative is to use ssh – secure shell.

Unix Filesystems

Just like Windows, Unix has to store files on disk and has a directory (folder) structure which shouldn't completely baffle someone who is confident navigating around using a Command Prompt under Windows. It also uses partitions although you shouldn't need to worry about this.

The key differences are

- the directory names are different!
- file and directory names are case sensitive
- a / (slash) is used instead of a \ (back slash) to separate directory names

Important Directories

dev – devices under Unix, including disks, input/output and other peripherals are treated as files which are kept in this directory

etc – contains system and some application configuration files

home – user home directories are kept here

usr – contains many different files for a range of purposes. /usr/bin contains system utilities such as *less*

var – variable length files such as log files are often kept here. /var/log/httpd may be used for web server log files, for example

Exactly which directories are used will depend on the particular version of Unix/Linux that you have and how the applications have been configured. For example, some applications may have their own etc and log directories.

Standard Input and Output

UNIX handles its command input and output as a sequence of characters with no internal structure; this is known as a stream. There are three streams:

- Standard Input - Input to a command, usually taken from the keyboard
- Standard Output - Output from a command, usually displayed on the screen
- Standard Error - Error messages, usually displayed on the screen

Standard output and standard error can be re-directed to file using a `>` (greater than / right arrow sign). Standard input can be re-directed to come from a file using a `<` (less than / left arrow sign).

Redirection and Piping

```
ls -l > filelist
```

Here the output of the `ls -l` command is redirected from standard output to a file called `filelist`.

To take the output of one command and use it as the input to another, use the pipe symbol `|` (which should be a vertical line with a gap in the middle, often in the bottom left corner of the keyboard above the `\` symbol).

```
[admin@gmulqube /etc]$ ls | more
BOOTED
DIR_COLORS
HOSTNAME
NET-CONFIG
X11
adjtime
admserv
aliases
aliases.db
aliases.master
aliases.old
aliases.rpmsave
analog.cfg
atalk
bashrc
build
cobalt-release
conf.modules
cron.d
cron.daily
cron.hourly
cron.monthly
cron.weekly
--More--
```

Useful Commands

This section is not intended to allow you to immediately start fully administering Unix systems. Instead it describes a number of the commands you may find useful for investigating a system either so that you can use them yourself or so that you are familiar with the basic terms if someone is trying to talk you through a problem over the phone for example.

Unlike (most) Windows commands, Unix commands and their options and arguments are case sensitive.

For most commands you can add `--help` to see basic online help.

`cd`

Change Directory

Unlike Windows, you **must** put a space between the `cd` and the name of the directory you want to change to

`chmod`

Change Access Mode

`chown`

Change Owner

`date`

Date

Display or set the current date. If you need to use this to set the date, use `date --help` to confirm the format you should use.

`df`

Disk Free

Shows the usage of the various disk partitions on the system including the amount of free space. By default the sizes will probably be shown in blocks, the `-h` option should show the sizes in a more readable format.

```
[admin@gmulqube admin]$ df -h
Filesystem      Size  Used Avail Capacity Mounted on
/dev/hda1       194M  117M   77M     60%   /
/dev/hda3       484M   6.9M  477M     1%   /var
/dev/hda4       5.0G   20M   5.0G     0%   /home
```

ls

List directory contents

```
[admin@gmulqube admin]$ ls
Network Trash Folder  index.html          private
[admin@gmulqube admin]$ ls -l
total 6
drwxr-xr-x   2 admin   users           1024 Jul 28  2002 Network Trash Folder
-rw-r--r--   1 admin   users           3592 Jul 28  2002 index.html
drwxr-x---   2 admin   users           1024 Jul 28  2002 private
[admin@gmulqube admin]$
```

As standard the ls command only shows the names of the files (including subdirectories) in the current directory (or another directory if specified as an argument). Sometimes the directories and different types of file may be shown in different colours although there will usually be an option to stop this.

man and help

These commands give access to different areas of online help. The help command is used for internal commands – those built in to the core of the system. The man command will usually be available and gives more detail about other commands if they have a man(ual) page available.

mkdir

Make Directory

Equivalent to the md command in Windows

more / less

These commands are useful if the output of a program would otherwise scroll off the top of the screen so you couldn't see it. They may also be used to view the content of a file. The difference between the two is that more will only let you progress forwards through a file. The less command will also allow you to scroll backwards.

In both cases pressing the spacebar will scroll a page forwards through the file, the cursor keys should scroll a line at a time and q will quit.

In less, use the up cursor key to go back up a line at a time or w to go up a page. There are many other options which may be view in the help, including alternative options in case the cursor keys do not work correctly.

passwd

Allows you to change the account password

pwd

Present Working Directory.

Most commands use the present working directory if no other argument is given. Therefore it's useful to be sure exactly which directory you're in before deleting files, for example. Here's an example of using pwd to see where you are whilst navigating around using cd

```
[admin@gmulqube admin]$ pwd
/home/httpd/html/users/admin
[admin@gmulqube admin]$ cd.. (This works on Windows)
sh: cd.: command not found
[admin@gmulqube admin]$ cd .. (Do this instead)
[admin@gmulqube users]$ pwd
/home/httpd/html/users
[admin@gmulqube users]$ cd /
[admin@gmulqube /]$ pwd
/
[admin@gmulqube /]$ cd ~
[admin@gmulqube admin]$ pwd
/home/httpd/html/users/admin
[admin@gmulqube admin]$
```

rm

Remove (file)

Similar to del in Windows. Use -r to recursively remove files and subdirectories



Do NOT type `rm * -r` without checking where you are first e.g. using `pwd` - many people have lost large chunks of their filesystem unexpectedly by doing just this...

rmdir

Remove Directory

su

Switch User

Allows you to log in as one user but to change to another one (including their group membership) without having to log out and in again. Can be useful if you just need to use one or two commands as a different user such as root, without being logged in with that much power all the time.

tail

View the tail end of a file. Use `tail -f` to watch the end of a file as new lines are added, idea for watching a log or error file grow.

uname

Display system information

```
[admin@gmulqube admin]$ uname --help
```

```
Usage: uname [OPTION]...
```

```
Print certain system information.  With no OPTION, same as -s.
```

```
-a, --all           print all information
-m, --machine       print the machine (hardware) type
-n, --nodename      print the machine's network node hostname
-r, --release       print the operating system release
-s, --sysname       print the operating system name
-p, --processor     print the host processor type
-v                 print the operating system version
--help             display this help and exit
--version          output version information and exit
```

```
Report bugs to sh-utils-bugs@gnu.ai.mit.edu
```

```
[admin@gmulqube admin]$
```

The Visual Editor

The vi text editor is commonly available on UNIX systems and although it's not exactly user friendly to start with, it can be quick to use and its wide availability make it worth learning the basics.

To start vi, call it with a filename:

```
$ vi httpd.conf
```

To start at a particular line, use **+linenumber** e.g.

```
$ vi +50 httpd.conf
```

To open the file at the end, use **+\$** e.g.

```
$ vi +$ httpd.conf
```

vi has three different modes.

- **Command Mode**
Command mode is used to edit and move around the text area. Other modes may only be entered by going through the command mode. This is the mode vi starts in.
- **Input Mode**
Input mode is where the user can insert text within the file being edited. There are a number of commands that will place you in this mode. The ESC key leaves input mode and returns you to command mode.
- **Last Line Mode**
Last line mode implements a further selection of commands. These are especially useful for handling ranges of lines. The last line mode is activated by entering : (colon) from the command mode.

vi returns to command mode after executing a last line command. Otherwise try the interrupt key CTRL+C (or sometimes the DEL/Break key).

Search and Replace

In command mode, use / (forward slash) to search forwards from the current position or ? to search backwards. Use n to find the next occurrence in the direction you were last going in.

In last line mode, use `:s/searchfor/replacewith/gc`

The `gc` at the end specifies that all occurrences on a line should be changed (`g`) and that confirmation should be given before each change (`c`) – both of these are optional and either or both may be omitted. To perform a search and replace on more than one line it is possible to specify the range of lines to be covered by using `:srange/searchfor/replacewith/`

Example:

```
:s2,100/old text/new text/g
```

This will replace any occurrence of “old text” with “new text” on lines 2 to 100 inclusive.

File commands

To save a file use `:w` – this will use the same name as the current file or use `:w filename` to save to a different file name.

If no changes have been made to the file then just use `:q` to quit, otherwise use `:wq` to save (write) the file and quit at the same time or `:q!` to quit without saving.